

# Реляційні бази даних, їхні об'єкти

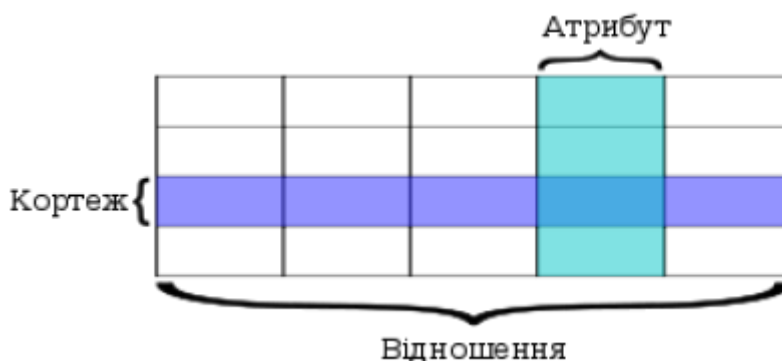
**Реляційна модель даних** — це спосіб структурування та організації даних в базі даних, що базується на концепції “реляційних таблиць”, де дані представлені у вигляді таблиць з рядками та стовпцями. Вперше була запропонована британським ученим співробітником компанії IBM Едгаром Франком Коддом (E. F. Codd) в 1970 році в статті «A Relational Model of Data for Large Shared Data Banks». В даний час ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні комерційні СУБД.

У реляційній моделі досягається набагато більш високий рівень абстракції даних, ніж в ієрархічній або мережевий. У згаданій статті Е. Ф. Кодда стверджується, що «реляційна модель надає засоби опису даних на основі тільки їх природної структури, тобто без потреби введення якої додаткової структури для цілей машинного представлення». Іншими словами, подання даних не залежить від способу їх фізичної організації. Це забезпечується за рахунок використання математичної теорії відносин (сама назва «реляційна» походить від англійського relation— «відношення»).

До складу реляційної моделі даних зазвичай включають теорію нормалізації. Крістофер Дейт визначив три складові частини реляційної моделі даних:

**Структурна частина** моделі визначає, що єдиною структурою даних є нормалізоване  $n$ -арне ставлення. Відносини зручно представляти у формі таблиць, де кожен рядок є кортеж, а кожен стовпець — атрибут, визначений на деякому домені. Даний неформальний підхід до поняття відносини дає більш звичну для розробників і користувачів форму представлення, де реляційна база даних являє собою кінцевий набір таблиць.

## Термінологія



**Маніпуляційна частина** моделі визначає два фундаментальних механізми маніпулювання даними — реляційна алгебра і реляційне числення. Основною функцією маніпуляційної частини реляційної моделі є забезпечення заходів реляційності будь-якої конкретної мови реляційних БД: мова називається реляційною, якщо вона має не меншу виразність і потужність, ніж реляційна алгебра або реляційне числення.

**Цілісна частина** моделі визначає вимоги цілісності сутностей і цілісності посилань. Перша вимога полягає в тому, що будь-який кортеж будь-якого відношення відмітний від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення має володіти первинним ключем. Вимога цілісності по посиланнях, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється в посиланому відношенні, у відношенні, на яку веде посилання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа повинно бути невизначеним (тобто ні на що не вказувати).



Едгар Франк «Тед» Кодд (23 серпня 1923 — 18 квітня 2003) — британський інформатик, який, працюючи у ІВМ, винайшов, окрім іншого, реляційну модель даних, теоретичну основу для реляційних баз даних.

Можна провести аналогію між елементами реляційної моделі даних і елементами моделі «сутність-зв'язок». Реляційні відносини відповідають наборам сутностей, а кортежі — сутностям. Тому, також як і в моделі «сутність-зв'язок» стовпці в таблиці, що представляє реляційне відношення, називають атрибутами.

Кожен атрибут визначений на домені, тому домен можна розглядати як безліч допустимих значень даного атрибуту. Кілька атрибутів одних відносин і навіть атрибути різних відносин можуть бути визначені на одному і тому ж домені.

Іменоване безліч пар «ім'я атрибута — ім'я домену» називається схемою відношення. Потужність цієї множини — називають ступенем чи «арністю» відносини. набір іменованих схем відносин представляє із себе схему бази даних.

Атрибут, значення якого однозначно ідентифікує кортежі, називається ключовим (або просто ключем). У нашому випадку ключем є атрибут «Табельний номер», оскільки його значення унікально для кожного працівника підприємства. Якщо кортежі ідентифікуються тільки зчепленням значень декількох атрибутів, то говорять, що відношення має складовий ключ. Ставлення може містити кілька ключів. Завжди один із ключів оголошується первинним, його значення не можуть обновлятися. Всі інші ключі відносини називаються можливими ключами.

На відміну від ієрархічної і мережної моделей даних в реляційній відсутнє поняття групових відносин. Для відображення асоціацій між кортежами різних відносин використовується дублювання їх ключів.

## Структура реляційної бази даних

Реляційна БД являє собою сукупність зв'язаних таблиць, що містять інформацію про об'єкти певного виду.

Рядок таблиці містить дані про один об'єкт (наприклад, про людину, автомобіль тощо), а стовпці таблиці містять різні характеристики цих об'єктів — атрибути (наприклад, прізвище, ім'я, дату народження, номер двигуна тощо).

У таблицях реляційної БД рядки називають записами, а стовпці — полями.

**Поля**

	№ п/п	Прізвище	Ім'я	День народження
+	1	Іванова	Оксана	12.12.1986
+	2	Петров	Максим	01.03.1987
+	3	Сидоров	Сергій	02.05.1986
+	4	Бабич	Вікторія	28.02.1987

Рис. 23.1

Поле реляційної БД (стовпець) містить дані одного типу. Записи (стовпці) містять сукупну інформацію про певний об'єкт.

Навіть порожня структурована БД містить певну інформацію про імена полів, опис типів даних тощо.

Структура реляційної БД визначає зв'язки між таблицями і можливі типи та властивості даних у них.

## Проектування реляційної бази даних

Проектування БД починається з визначення її структури за такими кроками:

- визначається перелік даних, які будуть зберігатися;
- визначається кількість і вміст таблиць для зберігання даних;
- визначаються для кожної з таблиць назви полів, їх тип, властивість тощо.

## Важливі аспекти реляційних БД


### SQL

SQL (Structured Query Language) – основний інтерфейс роботи з реляційними базами даних. SQL став стандартом Національного інституту стандартів США (ANSI) у 1986 році. Стандарт ANSI SQL підтримується всіма популярними ядрами реляційних баз даних. Деякі ядра також включають розширення стандарту ANSI SQL, що підтримують специфічний для цих ядер функціонал. SQL використовується для додавання, оновлення та видалення рядків даних, вилучення наборів даних для обробки транзакцій та аналітичних додатків, а також для керування всіма аспектами роботи бази даних.

### Транзакції

Транзакція в базі даних – це один або кілька операторів SQL, виконаних у вигляді послідовності операцій, що є єдиним логічним завданням. Транзакція є неподільна дія, тобто вона повинна бути виконана як єдине ціле і/або повинна бути записана в базу даних цілком, або не повинен бути записаний жоден з її компонентів. У термінології реляційних баз даних транзакція завершується або дією COMMIT, або ROLLBACK. Кожна транзакція розглядається як внутрішньо зв'язний, надійний та незалежний від інших транзакцій елемент.

### Відповідність вимогам ACID

Для дотримання цілісності даних усі транзакції у БД повинні відповідати вимогам  **ACID**, тобто бути атомарними, одноманітними, ізольованими та надійними.

**Атомарність** – це умова при якій транзакція успішно виконується цілком або, якщо якась із її частин не виконується, вся транзакція скасовується. **Одноманітність** – це умова за якої дані, записувані до бази даних у межах транзакції, повинні відповідати всім правилам і обмеженням, включаючи обмеження цілісності, каскади і тригери. **Ізольованість** необхідна для контролю над узгодженістю і гарантує базову незалежність кожної транзакції. **Надійність** передбачає, що всі внесені до бази даних зміни на момент успішного завершення транзакції вважаються постійними.

**ACID** (англ. **A**tomicity, **C**onsistency, **I**solation, **D**urability) — це набір властивостей, що гарантують надійну роботу транзакцій бази даних: атомарність, узгодженість, ізольованість, довговічність. В контексті баз даних, послідовність операцій з базою даних, яка задовольняє властивостям ACID, можна розглядати як одну логічну операцію над даними. Така послідовність операцій називається транзакцією. Наприклад, переказ коштів з одного банківського рахунку на інший містить численні операції, але є єдиною транзакцією.

1983 року Андреа Рейтер і Тео Хардер ввели акронім ACID, ґрунтуючись на вимогах, які сформулював раніше науковець Джим Грей.

### Переваги реляційної моделі:

- простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
- строгі правила проектування, які базуються на математичному апараті;
- повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
- для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

### Недоліки реляційної моделі:

- далеко не завжди предметна область може бути представлена у вигляді «таблиць»;
- в результаті логічного проектування з'являється безліч «таблиць». Це призводить до труднощів розуміння структури даних;
- БД займає відносно багато зовнішньої пам'яті;
- відносно низька швидкість доступу до даних.

## Джерела

- [Бази даних.СУБД Access](#)
- [uabooks.top](#)
- [aws.amazon.com](#)

From:

<https://library.vpuhluhiv.com.ua/> - **Вікі Глухівського ВПУ**

Permanent link:

[https://library.vpuhluhiv.com.ua/subjects:basic:informatika:base:reliasiini\\_bazy\\_danyh\\_ihni\\_objekty](https://library.vpuhluhiv.com.ua/subjects:basic:informatika:base:reliasiini_bazy_danyh_ihni_objekty)

Last update: **10.09.2023 21:09**

